

ON THE COMPUTATIONAL COMPLEXITY OF A GAME OF COPS AND ROBBERS

MARCELLO MAMINO

ABSTRACT. We study the computational complexity of a perfect-information two-player game proposed by Aigner and Fromme in [AF-84]. The game takes place on an undirected graph where n simultaneously moving cops attempt to capture a single robber, all moving at the same speed. The players are allowed to pick their starting positions at the first move. The question of the computational complexity of deciding what player has a winning strategy in an instance of this game was raised in the '90s by Goldstein and Reingold [GR-95]. We prove that the game is hard for PSPACE.

1. INTRODUCTION

We consider a two-player perfect-information game, COPS AND ROBBER (C&R), in which a given number n of cops attempts to capture a single robber by moving over the edges of an undirected graph. At the first move, the player controlling the cops chooses starting vertices for the n cops, then the robber's player chooses his vertex. After that, the cops and the robber are moved in alternating turns from one vertex to an adjacent vertex, all the cops move simultaneously. The game ends when one of the cops captures the robber. We prove in Theorem 2.1 that the problem of determining which player has a winning strategy in C&R is hard for PSPACE.

C&R has been first considered by Aigner and Fromme in [AF-84] generalizing a game that was proposed by Nowakowski and Winkler in [NW-83] and independently by Quilliot in [Qui-83]. Since then, this game has been the object of intense study from the combinatorial point of view, for a survey see [FT-08]. From the point of view of computational complexity, the first study appears in [GR-95], where Goldstein and Reingold prove that versions of the game played on directed graphs or starting from a fixed initial position are complete for EXPTIME. In [GR-95], Goldstein and Reingold first raise the question of the complexity of the unrestricted game.

Although the complexity of many similar games has been determined (the reader is referred again to [FT-08] for bibliography), the only lower bound for the game with elective initial positions on an undirected graph appeared only recently (for some positive algorithmic results see [BI-93, GR-95, HM-06]). Namely, in [FGK⁺10] it is

CMAF UNIVERSIDADE DE LISBOA, AV. PROF. GAMA PINTO 2, 1649-003 LISBOA, PORTUGAL

E-mail address: mamino@ptmat.fc.ul.pt.

Date: 27-II-2012.

2000 *Mathematics Subject Classification.* 91A24, 91A43, 68Q17.

Key words and phrases. computational complexity, cops and robber, pursuit game.

Partially supported by Fundação para a Ciência e a Tecnologia grant SFRH/BPD/73859/2010.

shown that the problem of determining whether the cops have a winning strategy in a given instance of C&R is hard for NP. As argued in [FGK⁺10], the difficulty in proving any complexity result for C&R lies in the extremely dynamic nature of the game. For example, the reader may observe that the cops can not really make any mistake (on a connected graph), since from whatever position they reach, they can go back to the initial position and restart the game. Arguably, this makes it very hard to force the cops to make any given move. As expected, the complexity of the game is much easier to assess when one adds further constraints on the set of possible moves, as in [GR-95] or [FGL-12]. In fact, we do precisely the same, reducing to C&R a new and more flexible pursuit game, which nevertheless is purely an extension of C&R—*i.e.* all instances of C&R are also instances of our game, see Section 3. Our game greatly simplifies the proof of existing results – see Corollary 3.2 – and also allows us to almost simulate C&R played on a directed graph, a game which is known to be EXPTIME-complete.

NP-hardness of C&R is proven in [FGK⁺10], and also in our Corollary 3.2, both proofs through DOMINATING-SET. Although the arguments are based on a slightly different concept, both produce a graph in which no real action takes place: if the robber is captured, capture occurs at the first move. In this case, the only difficulty for the cops lies in the choice of their initial position, and the game is indeed trivial for the robber. So, apparently, to the researcher trying to prove NP-hardness, the elective initial positions are more of an asset than a hindrance. However, as soon as one tries to attain hardness for higher complexity classes, the game can not be made trivial for the robber any more, and the initial choice of the cops is not hard enough to be exploitable. On the contrary, in this situation, the unpredictability of the initial positions becomes the main obstacle, and a different technique must be used.

In [GR-95] and [FGL-12], PSPACE- and EXPTIME-hardness results for variants of C&R are proven by reduction of games played on boolean formulæ. The typical method is to simulate operations on boolean variables by the action of several gadgets, which the robber is forced to traverse in a well determined order during his flight. At the end of the simulated boolean game, the robber has a chance to reach perpetual safety. The most immediate type of perpetual safety comes in the form of a safe subgraph (a subgraph in which the cops can never capture the robber, whatever the initial position), however the presence of safe subgraphs is incompatible with the players choosing their own initial positions, since the robber could just pick his vertex in it. In [FGL-12] is discussed a variant of C&R *without recharging*, *i.e.* imposing a constraint on the maximal number of moves that each cop can make, and, in the same paper, PSPACE-completeness is proven for this variant. Here, the uncertainty over the initial position is dealt with observing that no cop can start too far away from any vertex of the graph, since at least one cop must be able to reach any vertex in case the robber shows up there, hence the positions of the cops can be forced by connecting long antennæ to the vertices where we want a cop to be placed. A complicated collection of gadgets can then be devised to constrain the whole initial position and simulate QBF. For directed graphs, the idea of [GR-95] is to start with a construction having fixed initial positions, in which the robber reaches a safe subgraph if he wins the simulated boolean game. Then, the constructed graph is modified by substituting the safe subgraph by a reset mechanism, which is safe just as long as the cops are not

precisely in their initial positions. At this time, the robber is forced to exit the reset mechanism and play the boolean game. The directness of the graph is needed to make sure that the robber can not re-enter the reset mechanism, going backwards from his starting position instead of completing the game.

Our technique is kin to the directed graph case. In particular, we reduce QBF to our extended variant of C&R, which in turn reduces to C&R. We are not able to really simulate directed graphs, however our construction produces a graph which behaves as if it were directed, in the sense that both players can go the wrong way, but they have nothing to gain by doing it. Our graph is divided into levels arranged in a circular fashion, so that each level is connected just to the adjacent ones. The pursuit takes place while players go around the circle: if a given formula is true then the robber can be captured before he completes a single lap, if it is false then he can keep circling forever. At diametrical opposite places on the circle, we have two reset mechanisms, that can be used by either player to synchronize with his opponent. The obstacle to proving EXPTIME-hardness (hence completeness) by the same method is evident from the description above: all the interesting action in our graph takes place in a polynomially bounded number of moves (actually sub-linearly bounded, if we take every simulation step into account).

The *number of cops* needed to capture the robber has received intense study due to the long-standing conjecture of Meyniel that $O(\sqrt{|VG|})$ cops suffice on the graph G —this conjecture is still open, and the best bound so far, which has been obtained very recently [LP-11, SS-11], is $|VG|^{2^{-(1+o(1))\sqrt{\log|VG|}}}$. On the other hand, the *number of turns* has been considered just recently [BGHK-09], obtaining, for some classes of graphs, upper and lower bounds on the length of the longest games which are linear in the number of vertices. Clearly, if the length of any given game of C&R, provided that it is a win for the cops and that it is played optimally, could be bounded by a polynomial in the size of the graph, then, as a corollary of Theorem 2.1, C&R would be complete for PSPACE (the same would follow from results in [FGL-12]). However, even though no super-linear lower bound on the length of the longest games is known to the author, we believe that our techniques may lead to a proof of completeness of C&R for EXPTIME. This would imply a super-polynomial lower bound on the length of the longest games.

Acknowledgements. The results in this paper first appeared in the author’s Master’s thesis [Mam-04], supervised by Alessandro Berarducci. The author would like to thank Alessandro Berarducci for suggesting the problem and reading a preliminary version of this paper. The author is also obliged to Eleonora Bardelli for comments on a preliminary version, and for encouraging him to write the paper in the first place.

2. STATEMENT OF THE RESULT & OUTLINE OF THE PROOF

C&R is the problem of determining which player has a winning strategy in a given instance of the following game. An instance of the game is described by an undirected graph G and a natural number n smaller than the number $|VG|$ of vertices of G . The game is played by two contenders, *Cops* and *Robber*, with perfect information, by moving tokens on the vertices of G . At the first move, Cops places n tokens, *the cops*, each on a vertex of the graph, multiple cops are allowed on the same vertex. Then Robber places a single token, *the robber*, on some vertex. After that, the players take turns at moving their tokens from vertex to vertex. At his turn, each player can move along an edge of the graph each of his tokens, or leave it unmoved—each token moves at most once per turn, all the cops can move simultaneously. Cops wins if at any time the robber is on the same vertex as one of the cops (the robber has been *captured*). Robber wins if the robber escapes perpetually—or, equivalently, Robber wins after $|VG|^{n+1}$ moves have been played, so the game is not actually infinite.

We will prove the following result.

Theorem 2.1. *C&R is hard for PSPACE.*

In the three sections that follow, we will prove Theorem 2.1. Precisely, in Section 3, we will define a pursuit game and show that the new game and C&R are mutually LOGSPACE-reducible. Then, in Section 4, we will describe a construction proving that satisfiability for boolean formulae with quantifiers is LOGSPACE-reducible to our new game assuming fixed initial positions of cops and robber. Finally, Section 5 will use the construction of Section 4 to conclude the proof of Theorem 2.1. Although we will try to be as formally correct as reasonable, we believe that some abuses of nomenclature actually improve readability. In particular, the words *cops* and *robber* are going to be used interchangeably for the tokens and for the vertices upon which they reside; the same symbol is going to denote an instance of the problem C&R, the game constituting that instance, and the graph whereon that game is played; &c.

3. COPS AND ROBBER-WITH-PROTECTION

COPS AND ROBBER-WITH-PROTECTION (C&Rp) is a variant of C&R played on an undirected graph whose edges are labelled as either *protected* or *unprotected*. The rules for placing and moving tokens are the same as in C&R, in particular tokens can move along any edge irrespective of its label. The winning condition, on the other hand, changes as follows: Cops wins when he moves a cop to the vertex occupied by the robber *through an unprotected edge*, Robber wins if the robber escapes perpetually.

In the following, we will need to draw diagrams of labelled graphs for C&Rp. Protected and unprotected edges will be represented by broken (---) and continuous (—) lines respectively. Also, we will call *unprotected vertex* a vertex connected to itself by an unprotected edge, a *protected vertex* is a vertex which is not unprotected. Protected and unprotected vertices will be represented by empty (○) and full (●) circles respectively. If a cop and the robber happen to reside on the

same vertex, with the cop about to move, then the cop can capture the robber if and only if that vertex is unprotected.

Observe that we can easily reduce C&R to C&R_P by simply declaring all edges and all vertices unprotected. The rest of this section will be devoted to prove the converse.

Lemma 3.1. *C&R_P is LOGSPACE-reducible to C&R.*

Proof. Let an instance of C&R_P be given by a labelled graph G and a number n of cops. We will show how to construct a graph G' such that the instance of C&R described by G' and n is a win for Cops if and only if the given instance of C&R_P is a win for Cops.

More precisely, let $\{v_1 \dots v_g\}$ be the vertices of G . Then the vertices of G' will be partitioned into g subsets $G'_1 \dots G'_g$ so that the game of C&R played on G' simulates the game of C&R_P played on G in the following sense.

- A - The projection $\pi: VG' \rightarrow VG$ mapping G'_i to v_i is a graph homomorphism (neglecting the labels).
- B - If the projection of the robber can reach in one move a vertex v of G which is not threatened (according to the rules of C&R_P) by any of the projections of the cops, then the robber can reach in one move a vertex in $\pi^{-1}(v)$ which is not threatened (according to the rules of C&R) by any cop in G' .
- C - If the projection of the robber can not escape capture on the next move by the projections of the cops in G (playing C&R_P), then the robber can not escape capture on the next move in G' (playing C&R).

Properties A–C imply our claim that G is a win for n cops in C&R_P if and only if G' is a win for n cops in C&R. In fact, assume that we have a perpetually escaping C&R_P-strategy for Robber in G . Then he can successfully play C&R in G' by applying his C&R_P-strategy to the projected game in G (cops and robber in G are on the projections of cops and robber in G'). Because of property A, all the moves played in the projected game are legal. Using property B, given a safe move in the projected game, which we are assuming he always has, Robber can produce a safe move in G' . On the other hand, if Cops has a C&R_P-strategy for G , then he can get a C&R-strategy for G' by the same technique of looking at the projected game, and applying property C.

Now we proceed with the construction of G' . First we need an auxiliary graph H enjoying the following property. *Robber has a winning strategy for the instance of C&R played by n cops on H . Moreover, the vertices of H are partitioned into g subsets $H_1 \dots H_g$ in such a way that, for any position of cops and robber in which the robber is not captured, and for any $i = 1 \dots g$, Robber has a winning move that translates his token into H_i .* In order to construct H , pick a prime number p such that $\max(2n, g) < \lfloor \sqrt{p} \rfloor$. To keep the complexity of the construction polynomial, observe that p can be taken $< 2(1 + \max(2n, g))^2$. Now let

$$H_i = \left\{ (a, b) \in \{0 \dots p-1\} \times \{0 \dots p-1\} \mid (i-1) \frac{p}{\lfloor \sqrt{p} \rfloor} \leq a < i \frac{p}{\lfloor \sqrt{p} \rfloor} \right\}$$

and put an edge of H between vertices (a, b) and (a', b') if $aa' \equiv b + b' \pmod{p}$. It is easy to see that any two distinct vertices of H have at most one common

neighbour (not necessarily a vertex is a neighbour of itself), hence the cops can pose a direct threat over at most $2n$ of the neighbours of the robber, since each cop controls at most its own vertex and a common neighbour. On the other hand, for any given i , the robber has at least $\lfloor \sqrt{p} \rfloor > 2n$ neighbours in H_i : precisely one for every a with $(i-1) \frac{p}{\lfloor \sqrt{p} \rfloor} \leq a < i \frac{p}{\lfloor \sqrt{p} \rfloor}$. The required property follows.

We can now define G' . The set of vertices of G' coincides with the set of the vertices of H , indeed we let $G'_i = H_i$. Two vertices v and w of G' are connected by an edge when either their projections $\pi(v)$ and $\pi(w)$ are connected by an unprotected edge of G , or there is an edge of H between v and w and the projections $\pi(v)$ and $\pi(w)$ either coincide or are connected by a protected edge of G .

Remains to prove that G' satisfies properties [A–C](#).

- [A](#) - Immediate from the definition.
- [B](#) - If the projection of the robber can reach a safe vertex v in G , then no projection of a cop is connected to v by an unprotected edge, hence those cops that can reach $\pi^{-1}(v)$ can do it just moving through edges of H . On the other hand, since the projection of the robber can reach v , then the robber can reach $\pi^{-1}(v)$, and it can do that moving through any edge in H connecting its current position to a vertex in $\pi^{-1}(v)$. Follows from the construction of H that the robber can reach a vertex of $\pi^{-1}(v)$ avoiding all immediate threats.
- [C](#) - If the projection of the robber is doomed to be captured at the next move, then all the vertices reachable from its current position in G are connected to the projection of a cop by an unprotected edge. By [A](#) the projection of the robber must reside on one of this vertices at the next move. However, let v be such a vertex of G : since it is connected to the projection of a cop by an unprotected edge of G , then all vertices in $\pi^{-1}(v)$ are connected to a cop by an edge of G' . Hence all vertices reachable by the robber are under the immediate threat of some cop.

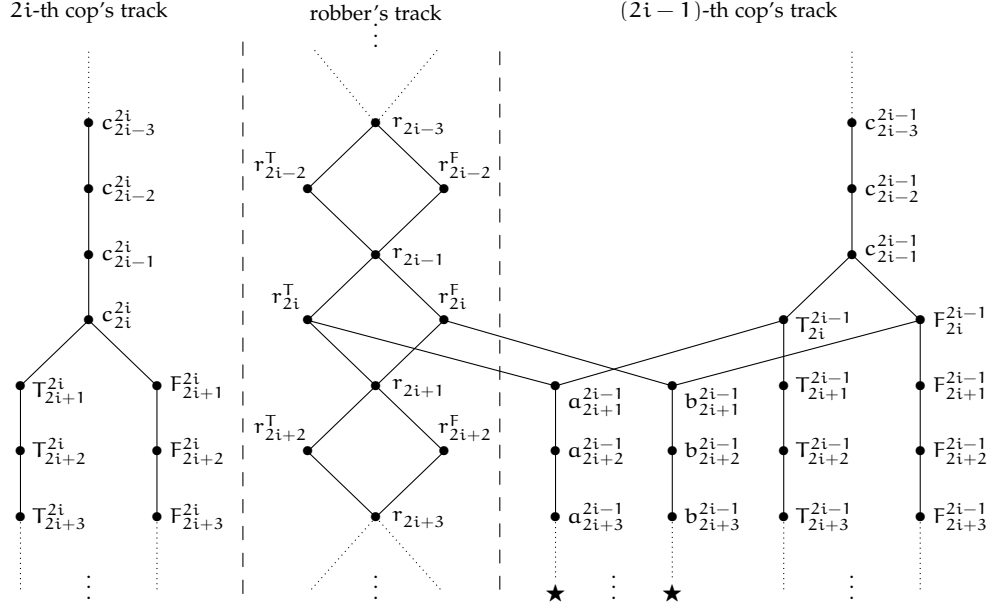
Finally, it is standard to verify that the construction of G' can be carried out in LOGSPACE. \square

The following corollary was first proven in [[FGL12](#)], and also it is, *a fortiori*, a consequence of our main result. Nevertheless, it provides an interesting example, since its proof is greatly simplified by the use of our game C&Rp.

Corollary 3.2. *C&R is NP-hard.*

Proof. By reduction of DOMINATING-SET. Let G be any graph, construct the labelled graph G' as follows. The set of vertices of G' coincides with the set of vertices of G . All vertices of G' are made unprotected. The edges of G' form a complete graph, those edges that are in G are labelled as unprotected, the others are protected.

We show that there is a dominating set of n vertices of G if and only if n cops can win on G' . In fact, if there is a dominating set, then the cops can be placed on that set at the first move, and wherever the robber shows up, it will be captured at the next move. If there is no dominating set, then at any move there is at least one unthreatened vertex, and the robber can be moved to it since G' is a complete graph. \square

FIGURE 1. Seven levels of three tracks of G_Φ

4. C&RP WITH A GIVEN STARTING POSITION

In this section we will prove the PSPACE-hardness of a simplified version of C&Rp where the starting position of all the tokens is fixed (instead of being decided by the players at their respective first moves). The result is uninteresting in itself, since C&R with given starting position is already known to be complete for EXPTIME [GR-95]. Nevertheless, in order to prove Theorem 2.1, we will exploit some peculiarity of the particular graph constructed below.

Let us define $C\&R^*$ as the problem of deciding what player has a winning strategy in our simplified game of C&Rp. More precisely an instance of $C\&R^*$ is given by a graph G whose edges are labelled as *protected* or *unprotected*, a natural number n , an n -element multiset of vertices of G for the Cops' tokens, and a distinguished vertex of G for the Robber's token. The problem is to determine whether Robber has a winning strategy for the game of C&Rp, played on G , starting with the tokens on the specified positions, Robber moves first.

Remark 4.1. When dealing with games, in common language, one often talks about how a game is played when both players follow *their best strategies*. However, the concept is not well defined, since apparently all moves are equally good for the player which has no winning strategy. Nevertheless, we will often say that such and such is going to happen when the game is *played correctly*, meaning that

- what constitutes a best strategy will be *fixed conventionally*, by explicit construction,
- and it will be proven that deviation from the stipulated best strategy will not afford any advantage, *i.e.* if any player has a winning strategy then the official strategy is a winning strategy for him.

Lemma 4.2. QBF is LOGSPACE-reducible to $C\&R^*$.

Proof. Let a quantified boolean formula be given

$$\Phi = \forall v_1 \exists v_2 \dots \forall v_{2n-1} \exists v_{2n} \phi(v_1 \dots v_{2n})$$

with $\phi(v_1 \dots v_{2n})$ quantifier free and in conjunctive normal form. We will construct a graph G_Φ , and show an appropriate initial position of cops and robber, such that Cops has a winning strategy for the game $C\&R^*$ if and only if Φ is true.

To make the construction clear, it is best to start by describing the *geography* of G_Φ , and only after that to go into the details. First of all, the reader should take a look at Figure 1, which depicts part of G_Φ . The graph G_Φ will be divided into two *stages*, each constituted by several *levels*—vertices aligned horizontally in Figure 1 are on the same level. Each level is connected just to the level above and to the level below, with no edge connecting a vertex of a level to another vertex of the same level.

The first stage lets Robber and Cops alternatively choose the value of the universally and existentially quantified variables respectively. These boolean values will be stored in the positions of $2n$ cops. This stage takes $2n$ levels, one for each of the variables. The first stage will be divided into *tracks* as well—tracks are represented vertically in Figure 1. Assuming correct play, we will have the robber and $2n$ cops starting at level 1, each on a separate track. Each of this tokens will move down one level per move, always remaining on *its* track. After move $2n$, all the tokens will be at level $2n + 1$, which is the first level of the second stage. At this level we will have two vertices for each of the variables, precisely one of which occupied by a cop, and a single vertex for the robber. The second stage will simply compute the value of ϕ . At the last level of the second stage we have a safe heaven: a protected vertex connected to the level above by a protected edge. If the robber reaches this vertex, it will never be captured.

Now to the description of the first stage. Figure 1 represents, in particular, part of this stage, and the reader is referred to it.

- The robber's track is constituted by a single vertex r_{2i+1} for each odd-numbered level, and two vertices r_{2i}^T and r_{2i}^F for each even-numbered level. All these vertices are unprotected. For all i , the vertices r_{2i}^T and r_{2i}^F are connected by unprotected edges to the vertex r_{2i-1} above and to the vertex r_{2i+1} below. The idea is that, while running down the track, the robber will determine the value of universally quantified variables by deciding through which side of each diamond to travel. Variable v_{2i-1} is assigned at Robber's $(2i - 1)$ -th move.
- Cops meant to store the values of existentially quantified variables run on a simple linear sequence of vertices, one per level, each connected to the one on the level below. Let c_j^{2i} denote the vertex on level j of the track assigned to variable v_{2i} . After level $2i$, the track of variable v_{2i} bifurcates into two linear sequences of vertices denoted T_j^{2i} and F_j^{2i} . The starting vertices T_{2i+1}^{2i} and F_{2i+1}^{2i} of these new sequences are both connected to c_{2i}^{2i} . Again, all vertices and all edges are unprotected. The bifurcations allow Cops to select the value of existentially quantified variables, precisely v_{2i} will be fixed at Cop's $2i$ -th move.
- Cops meant for universally quantified variables run on similar tracks. In particular, the track assigned to v_{2i-1} bifurcates after level $2i - 1$. However, the branch taken by the cop assigned to each of these tracks at the bifurcation must be determined

by the position of the robber. To this aim, we connect T_{2j}^{2j-1} and r_{2j}^T , through unprotected edges, to an unprotected vertex a_{2j+1}^{2j-1} placed at level $2j+1$, and in turn this vertex to a safe heaven. This way, if, after Robber's move $2j-1$, the robber is in r_{2j}^T , then Cops, who plays his own $(2j-1)$ -th move after Robber's one, is forced to move the cop which is in c_{2i-1}^{2j-1} to T_{2j}^{2j-1} , otherwise nothing would stop the robber from reaching the safe heaven through a_{2j+1}^{2j-1} . Because of how we are going to use the graph G_Φ in the proof of Theorem 2.1, we need to place the safe heaven at the end of a linear sequence of vertices spanning all the levels of G_Φ : this detail is immaterial for the proof at hand. Finally, we construct a similar device for vertices F_{2j}^{2j-1} and r_{2j}^F , connecting them to b_{2j+1}^{2j-1} .

The second stage begins at level $2n+1$. At this level we have two vertices T_{2n+1}^i and F_{2n+1}^i for each variable v_i , plus one vertex r_{2n+1} , and a few more of the vertices denoted by a and b . As explained, the a and b vertices, now, can be neglected. Assuming correct play, just before move $2n+1$, we have the robber in r_{2n+1} and precisely one cop in each pair of vertices T_{2n+1}^i and F_{2n+1}^i : the presence of this cop in T_{2n+1}^i denotes truth of v_i , the presence of the cop in F_{2n+1}^i denotes falsity. At level $2n+2$ we place one unprotected vertex for each clause of Φ . The vertex associated to clause c will be connected by an unprotected edge to T_{2n+1}^i whenever v_i is in c , and to F_{2n+1}^i whenever $\neg v_i$ is in c . All clauses are connected by unprotected edges to r_{2n+1} . Clearly, at move $2n+1$, the robber can be moved safely to one of the clauses' vertices only if that clause is false. Finally we connect all clauses to a safe heaven, so that the robber can reach it if and only if the formula is false.

To complete the construction of G_Φ , we attach a single unprotected vertex c_0 , by an unprotected edge, to r_1 . That vertex is on level 0, and we place two cops on it at the start of the game.

Now we prove that if Φ is true, then Cops has a winning strategy. First of all, observe that Cops can force the robber to go down its track, one level per move, until it reaches the second stage. To do so, he will use the two cops initially placed in c_0 to completely occupy the robber's track precisely one level behind the robber, so that Robber has to move his token one level down each move in order to avoid capture. The only way out of the track is through vertices named with a and b , so Cops will act as described above in order to block this escape. By following this strategy, Cops lets Robber choose the value of universally quantified variables at odd-numbered moves, and he can choose the value of existentially quantified variables at even-numbered moves. Observe that the order of choices coincides with the order of the quantifiers. Since Φ is true, Cops can make his choices so that $\phi(v_1 \dots v_{2n})$ is true. Hence the robber will be captured as soon as it enters one of the clauses.

Conversely, we prove that if Φ is false, then Robber has a winning strategy. Our strategy will move the robber down one level per move, hence no matter how they move, the two cops initially placed in c_0 will never be able to capture it. By the same reason, no cop moving backwards, or standing for one move, can capture the robber, so we can assume that all cops move down one level per move, and, in particular, none of them can leave its track. By the usual reason, Robber can force the value of odd-numbered variables. Now, by our assumption, Cops has chosen

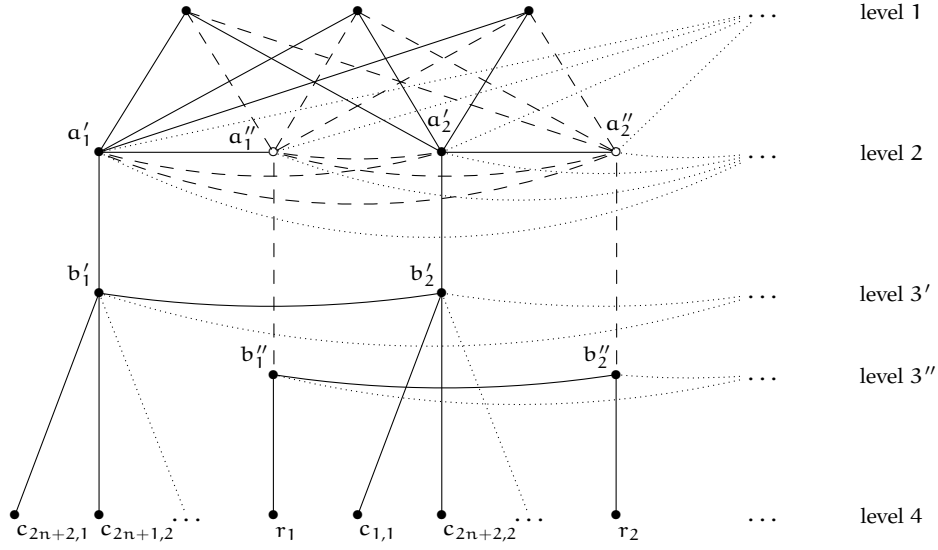


FIGURE 2. A portion of a reset mechanism

values to even-numbered variables at proper times. However, since Φ is false, Robber can make his choices so that $\phi(v_1 \dots v_{2n})$ will be false, hence he will be able to move the robber to an unthreatened clause, whence it will reach safety. \square

5. PROOF OF THE MAIN THEOREM

In this section we will prove Theorem 2.1. Our technique is, again, by reduction of QBF to C&RP. In particular, we will connect a few copies of the graph G_Φ constructed in the previous section to two *reset mechanisms*—a portion of one of which is shown in Figure 2. The function of the reset mechanism is to substitute the safe heaven at the end of G_Φ and to allow either player to force the initial position. As we will see, the mechanisms have been devised so that the robber can safely inhabit either of them unless all the cops are employed to chase it through a very specific maneuver, and doing that the initial position of the proof of Lemma 4.2 is attained. If Φ happens to be false, then Robber can move his token safely through G_Φ to the other reset mechanism. On the other hand, we will see that Cops as well has means to force Robber to get into the starting position. So Cops will have a winning strategy in our instance of C&RP if and only if he has a strategy for C&RP * on the graph G_Φ if and only if Φ is true.

The argument that follows and the proof of Lemma 4.2 are actually one single proof. We decided to separate a substantial portion of it into Lemma 4.2 in order to give a more orderly exposition. Nevertheless the reader will not understand the rest of this section unless he diligently went through Section 4 before.

Let us assume that we have formula

$$\Phi = \forall v_1 \exists v_2 \dots \forall v_{2n-1} \exists v_{2n} \phi(v_1 \dots v_{2n})$$

as in the proof of Lemma 4.2, we are going to construct a labelled graph G such that Cops has a winning strategy for $2n+2$ cops if and only if Φ is true. We

will use $4n + 4$ slightly modified copies of the graph G_Φ connected to two reset mechanisms. It is convenient to give an overview of how these components fit together to make the graph G . The copies of G_Φ are modified by removing the vertex c_0 and all the safe heavens. So the top level of each copy of G_Φ is constituted by the starting vertices of the robber and all the cops, and the bottom level is constituted by the vertices representing the clauses of ϕ and vertices a_{2n+2}^{odd} and b_{2n+2}^{odd} . For clarity's sake, the reset mechanisms are divided into levels as well—as usual, vertices aligned horizontally in Figure 2 are on the same level. Broadly speaking, we will arrange $2n + 2$ copies of G_Φ so that their top levels coincide with the bottom level of one of the reset mechanisms; the top level of the other reset mechanism will coincide with the bottom levels of these copies of G_Φ ; then the top level of the other $2n + 2$ copies of G_Φ will coincide with the bottom level of this second reset mechanism; and, to close the circle, the bottom level of the last $2n + 2$ copies of G_Φ is going to coincide with the top level of the first reset mechanism.

More precisely, fix $2n + 2$ of our modified copies of G_Φ . One reset mechanism is constructed as follows.

- On level 1, we place all the vertices belonging to the bottom levels of the $2n + 2$ copies of G_Φ , these vertices include those representing the clauses of ϕ .
- On level 2, we place $2n + 2$ unprotected vertices $a'_1 \dots a'_{2n+2}$ and $2n + 2$ protected vertices $a''_1 \dots a''_{2n+2}$. For each $i = 1 \dots 2n + 2$ we put an unprotected edge between a'_i and a''_i , unprotected edges between a'_i and all vertices of level 1, and protected edges between a''_i and all vertices of level 1. Finally, all the vertices of level 2 are connected together in a complete graph of protected edges.
- On level 3' we put vertices $b'_1 \dots b'_{2n+2}$. Each b'_i is connected by an unprotected edge to the corresponding a'_i . All vertices of level 3' are connected together in a complete graph of unprotected edges.
- On level 3'' we put vertices $b''_1 \dots b''_{2n+2}$. Each b''_i is connected by a protected edge to the corresponding a''_i . All vertices of level 3'' are connected together in a complete graph of unprotected edges.
- Finally, level 4 is constituted by the top levels of the other $2n + 2$ copies of G_Φ arranged as follows. Let $G_{\Phi,1} \dots G_{\Phi,2n+2}$ be our copies of G_Φ . Call r_i the starting vertex for the robber in $G_{\Phi,i}$ and call $c_{i,1} \dots c_{i,2n}$ its cops' starting vertices. For all i , we connect r_i to b'_i with an unprotected edge, and $c_{i,1} \dots c_{i,2n}$ to $b'_{i+1} \dots b'_{2n+2} b'_1 \dots b'_{i-2}$, again with unprotected edges.

The second reset mechanism is constructed symmetrically.

Now we show a winning strategy for Robber assuming that Φ is false. Observe that each cop can pose a threat on at most one of the vertices a''_i , so, after Cops has placed his tokens, there is at least one unthreatened a''_i in one of the reset mechanisms. The robber should be placed in that vertex. Now, let's focus on the reset mechanism in which the robber has been placed. Until all the cops occupy precisely all the vertices a'_i of that reset mechanism, the robber will simply stay on an unthreatened a''_i —it can be moved between them through the complete graph of protected edges. When all the cops occupy all the vertices a'_i , Robber will have his token in, say, a''_k . Now he will send it to b''_k , whence it will enter the k -th copy of G_Φ connected to that reset mechanism, and arguably emerge unscathed from it in the opposite reset mechanism. To prove that Robber's plan actually works observe that

- No cop can capture the robber while it moves from a_k'' to b_k'' to r_k .
- The robber is travelling along a geodesic toward level 2 of the opposite reset mechanism, hence no cop will arrive there before it, not even passing through another copy of G_Φ or going backwards (it is here that we use the fact that the linear sequences of vertices indicated by letters a and b in Figure 1 are no longer nor shorter than the *normal* path to safety).
- By the time the robber is in r_k , precisely $2n$ cops can reach their $2n$ starting vertices in the copy of G_Φ that the robber just entered. Hence, following the strategy detailed in the proof of Lemma 4.2, the robber will reach one of the clauses of ϕ , escape capture there having chosen the values of the variables properly, and finally move to level 2 of the new reset mechanism before any cop can be there.

Remains to be proven that if Φ is true, then Cops has a winning strategy. In the following, we will assume that ϕ has at least 8 variables (we need to have at least 9 cops around), and that ϕ has at least one non-empty clause (we need G to be connected). Clearly, this goes without loss of generality. The intermediate goal of Cops is to reach the following position, with the Robber about to move:

- robber in some a_i'' of one reset mechanism,
- $2n + 1$ of the cops in vertices a_j' of the same reset mechanism, with $j \neq i$,
- and the remaining cop either in a_i' or in a_i'' .

To this aim, he places initially three cops in vertices b_1' , b_2'' , and a_3' of each reset mechanism (6 cops total), we don't care where the remaining cops are placed. The cops at a_3' make a barrier at level 1 of their respective reset mechanisms, and the cops at b_1' and b_2'' make a barrier at level 3. Hence, as long as these 6 cops stay in place, the graph is effectively divided, from Robber's point of view, into $4n + 6$ disjoint components: the $4n + 4$ copies of G_Φ and the two reset mechanisms. We claim that if Robber places his token in one of the copies of G_Φ , then Cops can win using three additional cops (over the 6 above). *The strategy is as follows. Two cops reach the robber's track in the copy of G_Φ inhabited by the robber, they can do that since G is connected, then they move downwards until they reach the same level as the robber (with the robber about to move). From this moment on, these two cops will be kept at the same level of the robber, i.e. they will be moved up or down whenever the robber is moved up or down, also they will be placed so that they occupy their level of the robber's track completely. Hence, from this moment on, the robber can not access the robber's track, which implies that it must be on a cop's track and it can not move from this cop's track to another. Since all cop's track are trees, a single additional cop is sufficient to capture the robber. Hence we know that Robber must place his token in one of the reset mechanisms.*

Now we assume that Robber placed the robber in one of the reset mechanisms. We will now explain how Cops can attain his intermediate goal. All the action described in this paragraph takes place inside the reset mechanism chosen by the robber. Cops has three tokens in vertices b_1' , b_2'' , and a_3' . He now moves all the other tokens to vertices $a_4' \dots a_{2n+2}'$. While doing that, the cops in b_1' , b_2'' , and a_3' are not moved (unless the robber tries to escape), so at the end of the maneuver the robber must be either in a_1'' , in a_2' , or in a_2'' : the only unthreatened vertices of the reset mechanism. Now Cops can reach his goal by one of the following sequences of moves, which are forced for Robber:

- If the robber is in a_1'' : cop from b_1' to a_1' – robber in a_2' or a_2'' – if the robber is in a_2'' : cop from b_2'' to a_2'' and we are done—otherwise the robber is in a_2' : cop

- from a'_1 to b'_1 and cop from b''_2 to a''_2 – robber either in a''_1 or a''_2 – cop from a''_2 to a'_2 and cop from b'_1 to a'_1 .
- If the robber is in a'_2 : cop from b''_2 to a''_2 – robber either in a''_1 or a''_2 – cop from a''_2 to a'_2 and cop from b'_1 to a'_1 .
 - If the robber is in a''_2 : cop from b'_1 to a'_1 and cop from b''_2 to a''_2 .

Finally, the robber is in some a''_j , and all the vertices a'_i are occupied by cops except at most a'_j , in which case its cop must be in a''_j . Without loss of generality we can assume $j = 2$. The only option for Robber is to move his token down to b''_2 .

Now the following happens:

- all the cops in $a'_3 \dots a'_{2n+2}$ are moved down to $b'_3 \dots b'_{2n+2}$,
- if a'_2 is occupied by a cop, then that cop is not moved, otherwise there is a cop in a''_2 , and that cop is moved to a'_2 ,
- the cop in a'_1 is moved to a''_1 .

At this point there are two cases, either the robber moves to r_2 or it moves to one of the vertices b''_i (here including the case if it stays in b''_2).

- In the first case, the cops in $b'_3 \dots b'_{2n+2}$ must be moved to $c_{2,1} \dots c_{2,2n}$, the cop in a'_2 goes to a''_2 , and the cop in a''_1 goes to b''_1 . This way the robber can not be moved back to b''_2 , and it can either be moved into the robber's track of the second (because it is in r_2) copy of G_Φ , or be left in r_2 . In either case the cops in a''_2 and b''_1 will be moved to b''_2 at the next move, forcing it downwards. These two cops have the same function of the cops in vertex c_0 of the proof of Lemma 4.2, although they may be lagging two levels behind the robber instead of one. The reader can check that the argument of that proof applies from now on to the second copy of G_Φ . Hence the robber can not emerge from that copy of G_Φ uncaptured.
- In the second case, let's assume that Robber moved his token to b''_j . Then Cops moves the cops from $b'_3 \dots b'_{2n+2}$ to $b'_{j+1} \dots b'_{2n+2} b'_1 \dots b'_{j-2}$, the cop in a'_2 is sent to a'_j , and the cop in a''_1 is sent to b''_1 . Now the robber is forced to r_j and the strategy of the first case applies.

To conclude the proof, suffices to verify that our graph G can be constructed in LOGSPACE, which is standard. \square

REFERENCES

- [AF–84] M. Aigner and M. Fromme, *A game of cops and robbers*, Discrete Appl. Math. **8** (1984), no. 1, 1–11. [MR 739593 \(85f:90124\)](#)
- [BI–93] Alessandro Berarducci and Benedetto Intrigila, *On the cop number of a graph*, Adv. in Appl. Math. **14** (1993), no. 4, 389–403. [MR 1246413 \(94g:05079\)](#)
- [BGHK–09] A. Bonato, P. Golovach, G. Hahn, and J. Kratochvíl, *The capture time of a graph*, Discrete Math. **309** (2009), no. 18, 5588–5595. [MR 2567962 \(2010j:05273\)](#)
- [FGK⁺10] Fedor V. Fomin, Petr A. Golovach, Jan Kratochvíl, Nicolas Nisse, and Karol Suchan, *Pursuing a fast robber on a graph*, Theoret. Comput. Sci. **411** (2010), no. 7–9, 1167–1181. [MR 2606052 \(2011f:68055\)](#)
- [FGL–12] Fedor Fomin, Petr Golovach, and Daniel Lokshtanov, *Cops and robber game without recharging*, Theory of Computing Systems **50** (2012), 611–620, 10.1007/s00224-011-9360-5.
- [FT–08] Fedor V. Fomin and Dimitrios M. Thilikos, *An annotated bibliography on guaranteed graph searching*, Theoret. Comput. Sci. **399** (2008), no. 3, 236–245. [MR 2419780 \(2010a:05176\)](#)
- [GR–95] Arthur S. Goldstein and Edward M. Reingold, *The complexity of pursuit on a graph*, Theoret. Comput. Sci. **143** (1995), no. 1, 93–112. [MR 1330675 \(96j:68091\)](#)
- [HM–06] Geňa Hahn and Gary MacGillivray, *A note on k-cop, l-robber games on graphs*, Discrete Math. **306** (2006), no. 19–20, 2492–2497. [MR 2261915 \(2007d:05069\)](#)
- [LP–11] Linyuan Lu and Xing Peng, *On meyniel’s conjecture of the cop number*, Journal of Graph Theory (2011), n/a–n/a.
- [Mam–04] Marcello Mamino, *Complessità computazionale di un gioco combinatorio sui grafi*, Master’s thesis, Università di Pisa, 2004.
- [NW–83] Richard Nowakowski and Peter Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math. **43** (1983), no. 2–3, 235–239. [MR 685631 \(84d:05138\)](#)
- [Qui–83] Alain Quilliot, *Discrete pursuit game*, Congr. Numer. **38** (1983), 227–241. [MR 703252](#)
- [SS–11] Alex Scott and Benny Sudakov, *A bound for the cops and robbers problem*, SIAM J. Discrete Math. **25** (2011), no. 3, 1438–1442. [MR 2837608](#)